Remarks:

Reconsideration of the above referenced application in view of the enclosed remarks is requested. Claims 2, 6, 7, 9, 14, 18, 19, 22-24, 27-29, 39 and 42 are amended. Pending Claims 1-37, 39 and 41-42 remain in the application.

ARGUMENT

Objection to Claims:

Claims 24 and 28 are amended to correct minor informalities, at the request of the Examiner. Thus, the objections are moot based on the above amendments.

Rejections under 35 U.S.C. § 101

The rejections under § 101 are believed moot based on the above amendments. Each claim is directed to a useful, tangible and concrete result. It is also believed that all claim elements and limitations are interconnected and do not appear to be disjoint, as seems to be suggested by the Examiner.

Rejections under 35 U.S.C. § 102

Claims 1, 27-28 are rejected under 35 U.S.C. § 102(e) as being unpatentable over USPN 6,622,252 to Klaasen et al. (hereinafter, "Klaasen et al."). This rejection is respectfully traversed and Claims 1, 27-28 are believed allowable based on the foregoing and follow discussion.

Klaasen et al. teach a system for selecting modes of a drive (2-speed device) based on whether a portable device is operating under battery power. The point of their invention is to provide a lower speed using less power when under battery power. Klaasen et al. do not teach or suggest that buffering data will enhance their invention, as admitted by the Examiner. At the cited reference, Col. 1, lines 40-57, Klaasen et al. teach disengaging the power saving mode prior to commencing read and write operations (lines 55-57). At the cited reference, Klaasen et al. teach a specific method that automatically spins the drive up to normal speed once the disk drive is initiated (lines 51-55). Klaasen et al. make no determination of the activation of the disk drive in order to select whether the write operation should be buffered. Klaasen et al. teach a method limited to the spinning up and down of the spindle to preserve battery power. Klaasen et al.

teach that if the spindle is not spun up to normal operating velocity when an access operation is "initiated," that the spindle speed is increased until the disk is rotated at the normal operating velocity "prior to beginning the read or write operation." The power savings mode is disengaged prior to the commencement of the operation.

The Examiner asserts that the buffering, as recited in Claims 1, 27-28 is "inherent in Klaasen et al. This assumption is incorrect. The Examiner misunderstands the use of the term "buffer." A *buffer* is widely understood in the art to be a separate storage location used to temporarily hold data. Data must be written into and read from. For instance, one common understanding of the term is found on the public Internet at URL searchsmb*techtarget*com/sDefinition/0,,sid44_gci211713,00.html where periods are replaced by asterisks in URLs in this document to prevent inadvertent hyperlinks. The definition is:

> "A buffer is a data area shared by hardware devices or program processes that operate at different speeds or with different sets of priorities. The buffer allows each device or process to operate without being held up by the other. In order for a buffer to be effective, the **size of the buffer and the algorithms for moving data into and out of the buffer need to be considered by the buffer designer**. Like a cache, a buffer is a "**midpoint holding place**" but exists not so much to accelerate the speed of an activity as to support the coordination of separate activities.
>
> This term is used both in programming and in hardware. In programming, buffering sometimes implies the need to screen data from its final intended place so that it can be edited or otherwise processed before being moved to a regular file or database." [emphasis added]

Another source, published by Microsoft Corp., on the public Internet at URL at www*Microsoft*com/technet/archive/wfw/7_agloss.mspx?mfr=true defines a buffer and buffering as follows:

buffering: The process of using buffers to hold data being moved to or from I/O devices such as serial ports and disk drives.

buffers: A reserved part of memory where data is held temporarily until the data is transferred from storage to another location in memory. Some printers have their own buffers.

It can be seen from these common definitions that a buffer is a separate holding area. Further, the term "buffering" is commonly understood to be the act of writing to and reading from a buffer. However, Klaasen et al. do not teach if the device is determined to be deactivated,

buffering the write operation to physical memory. A mere determination of activation and inactivation is not enough to make adding buffering obvious. Nor is it enough to make buffering data to physical memory on that condition (inactivation) obvious. In fact, if anything is "inherent" in Klassen et al. it is that the access operation is held up until the spindle is at normal velocity. The Examiner admits that the operation is temporarily held – but it is held in place. Whether the access operation is held in "physical memory" or not is irrelevant, because no buffering into physical memory has taken place. The access operation stays where it is. There is no teaching or suggestion that the access operation is moved, or buffered, to a separate memory area, or *buffer*. One can analogize this situation to a people standing in line for a ride at an amusement park. When the ride is running, one waits in line. When the people are allowed to enter the ride area, they begin to move. However, when the ride is full, the line stops. The people are temporarily held in place, in line. They are not moved to a completely separate holding area – they are temporarily held until the line begins to move again. This is the only logical assumption that can be made regarding the teachings of Klaasen et al., i.e., when the spindle is not yet at normal velocity, the access operation is held up. But nowhere is there a teaching or suggestion that the access operation be moved to a separate holding area, or buffer. This action cannot be assumed, and it is not inherent in the teachings. Thus, the Examiner's assertion cannot be logically, or legally maintained.

Klaasen et al. does not show each and every limitation of Applicant's recited claims, as required by a rejection under § 102. Thus, the Examiner has failed to provide *prima facie* evidence of anticipation and Claims 1 and 27-28 must be allowed to issue at the earliest possible time.

Claims 31-32 and 35-37 are rejected under 35 U.S.C. § 102(e) as being anticipated by USPN 6,647,499 to Morcom (hereinafter, "Morcom"). This rejection is respectfully traversed and Claims 31-32 and 35-37 are believed allowable based on the foregoing and following discussion.

As for Claim 31, the Examiner asserts that Morcom teaches determining the power state of a device. Morcom teaches powering on a storage device and then copying a maximum amount of data to cache. Immediately following the read, the storage device is powered down.

In contrast, Applicant's claimed invention determines the current power state of the device and then based on that state, selectively reading a superset of the requested file portion into physical memory. Morcom reads data until the cache is full, but there is no requirement that the data is related to the requested portion, nor does Morcom teach that a lesser amount of data be read. This teaches away from selectively reading a superset. It is inherent in the limitation of selectively that a superset might not be read in some cases, or that the superset size may differ. Morcom makes no selection. Data is read until the cache is full. Period. No selection for a subset, superset or any other type of logically related information is made. Thus, this limitation is not taught or suggested by Morcom. An important limitation that is not taught or suggested by Morcom is selectively reading a superset. Morcom reads data until the cache is full. There is no selection determination at all. The amount of data read by Morcom is limited by the size of the cache only. Morcom does not make an intelligent decision, or any decision at all, how much data to read. Therefore, Morcom teaches away from selecting the amount of data to read. This selection is not taught or suggested by Morcom. Morcom teaches always filling the cache, not selectively filling the cache. Further, as discussed above, Morcom does not teach that the data is logically related to the requested portion, but only that data, related or unrelated is read until the caches is full. There is no distinction at all as to the relatedness of the data at the cited reference. Thus, the Examiner fails to meet the requirements of a § 102 rejection and present a *prima facie* case of anticipation. Since the cited reference fails to show each and every limitation of the claim and Claim 31 and its progeny are believed allowable.

As for Claim 32, Morcom does not teach that a superset is the entire file. Morcom teaches reading data until the cache is full. If the cache is smaller than the file, then Morcom cannot copy the entire file, which is in contrast to Applicant's claim. The Examiner has argued that no limit is placed on the size of the cache. However, it will be apparent to one of skill in the art that any type of cache will be finite in size and there is no way to limit the size of a file. Therefore, the Examiner's assertion is faulty. Claim 32 requires that the superset is selectively read and the selection is the entire file, which is logically related to the requested data. Further, if Morcom's cache is excessively large, superfluous and unnecessary (and unrelated) data will be read just to fill the cache. This will put an unnecessary strain on the battery just to fill the cache. There is no selection or decision being made by Morcom to affirmatively choose to read the

entire file. Thus, Morcom's method will not result in Applicant's invention. The limitations of Claim 32 are not taught or suggested by Morcom. Therefore, Claim 32 and its progeny are believed allowable.

As for Claim 35, the Examiner asserts that Morcom teaches *wherein the requested file portion is read from the device and returned to a requesting process before a remainder of the superset is read into physical memory.* Morcom does not teach that the requested portion is returned to the requesting process before a remainder is read into physical memory. As discussed above, Morcom does not teach a superset as defined by Applicant, and Morcom does not teach the temporal requirements of returning the requested portion first. Morcom merely teaches reading as much data as will fit in the cache. The Examiner cites Col. 5, lines 7-26. However, here, Morcom teaches reading data from the cache and if more data is *required,* then returning to block 203 to request more data for processing. This process teaches that requested data is read from the cache first and if all of the requested data is not in the cache, then reading more data from the device.

In contrast, Applicant's invention requires that the requested (required) portion of data is read from the device (not cache memory) and returned to the requesting process. Only then is the superset (i.e., not requested, but logically related data) read into physical memory. These two processes are not similar, and provide a different result. Morcom teaches retrieving the requested data, and Applicant requires that a superset of the required (requested) data is read into cache (physical memory) and not that the data is read from cache. Therefore, Claim 35 and is believed allowable.

Claims 36-37 are believed allowable based on being dependent on an allowable claim, as discussed above. The Examiner has failed to provide *prima facie* evidence of anticipation for the limitations of *selectively reading a superset of the requested file portion from the device into physical memory.* Claims 36 and 37 rely on the combination of the elements of the parent claims and cannot be viewed in a vacuum. While references may show accessing data from a buffer or deactivating a device after a read, the reference fails to teach the "superset" as recited in Claim 31 or the selective aspect of determining what the superset should comprise. Therefore, Claims 36-37 are believed allowable.

Claims 39 and 42 are rejected under 35 U.S.C. § 102(b) as being anticipated by USPN 5,812,883 to Rao (hereinafter, "Rao"). This rejection is respectfully traversed and Claims 39 and 42 are believed allowable based on the foregoing and following discussion.

As for Claim 39, the Examiner likens Rao's SCSI controller board to an intermediate file system driver providing read/write policy to the file system driver. The Examiner also likens Rao's drive controller board to a file system driver, as defined and claimed by Applicants. These assertions are in error. Specifically, Rao's drive controller board 206 may only accept rudimentary or low level commands which "pertain to the physical operation of the disk drive including positioning of the disk drive head and electrical signals to the disk drive head used to read or write information, for example. Other commands which do not directly pertain to the rudimentary control and positioning of the disk heads, disk speed, etc., are considered high level commands and are processed by the SCSI controller board 202." Even if the Examiner's characterization of the SCSI controller board and drive controller board were correct, Rao does not teach or suggest the limitations recited in the claim. The Examiner misunderstands the purpose of the intermediate FSD. Claim 39 clearly recites that *an intermediate file system driver to receive user customized parameters and to receive file system requests, the intermediate file system driver to determine read/write policy for controlling access to the file system driver based on the user customized parameters.* Rao teaches that the customizable parameters are received by the SCSI controller board and then transferred to the storage drive controller board 206, which the Examiner likens to the FSD (Col. 7, lines 18-22). Thus, only those parameters affecting the operation of the drive controller board may be input.

In contrast, Applicant's describe a system where the FSD is the driver performing read/write of the device. All policy and selective buffering decisions determined and performed in the intermediate FSD. Applicant's claim requires that the read/write policy determines the policy used to control the FSD, based on user customized parameters. The drive controller board of Rao cannot accept this type of information, as by definition, it only accepts low level commands. The erasable NV memory on the controller board does not determine selective buffering, but may determine an inactivity threshold that will cause the drive to spin down. Only hardware related parameters are described. In contrast, Applicant describes the read/write policy

to be defined as including identifying which programs are registered, etc. Paragraph [0019] of the Specification defines the parameters as:

> "...parameters or characteristics of the selective buffering scheme described here. Such parameters may include items such as which applications or processes are to participate in, or be excluded from, the buffering scheme, buffer size, time-out periods, and/or specific conditions under which the selective buffering should be employed or bypassed. For example, the user customization interface 108 may enable the user to specify circumstances under which a file write request will not be buffered but rather will force a device access to commit the data to non-volatile storage. These conditions may include, for example, an occurrence of a user's explicitly choosing the "save" command from the application's menu structure, a specified percentage of the document being changed, a specified volume of data being buffered, a specified number of writes being buffered, the battery's power level reaching a specified threshold, and/or the passage of a specified amount of time since the last committing of the buffer's contents to non-volatile storage. In general, the user customization interface 108 may allow a user to override any settings, defaults or other conditions that may have come about as a result of interactions among the various software components in the system (e.g., operating system, application programs, utilities, etc.).

Rao does not teach the type of parameters as defined by Applicant. Regardless of the structural differences, Rao fails to teach or suggest the limitation of *selectively* buffering write requests to physical memory. The Examiner has previously admitted that Rao fails to teach this limitation on page 24 (item 109) of the Office Action dated Oct. 2, 2006. It is improper for the Examiner to admit that one of the limitations is not taught by the reference, and to maintain the rejection. Therefore, this rejection is improper and must be withdrawn. The selectively buffering is an important limitation and cannot be glossed over by the Examiner. The Specification clearly describes how and why these selections are made. Thus, the term has specific meaning in the context of the disclosure and must be interpreted as described.

Further, the Examiner asserts that Rao teaches that prolonged period of inactivity would minimize activation and deactivation of the device by being "always on." This state is contrary to a purpose of Applicant's invention which is to also minimize power consumption. Thus, Applicant amends Claim 39 to more clearly recite *wherein the read/write policy is to maximize power saving while minimizing ~~minimize~~ at least one of (a) unnecessary device access operations and (b) unnecessary device activation-deactivation operations*. As such it will be understood that the minimization is done to conserve power and not just to stay always on or off.

Rao does not teach or suggest this limitation and does not result in Applicant's invention or provide the same advantages.

Moreover, Claim 39 is amended to more clearly recite that user customized parameters identify whether a process is exempt from buffering, i.e., requires a write to the device in conventional manner. This read/write policy for processes is not taught or suggested by the cited reference. Thus, Claim 39 and its progeny are believed allowable as amended.

As for Claim 42, Rao teaches using a control panel-like method to select the model and manufacturer of a disk drive and change the operating parameters of the actual drive. Rao does not teach selective buffering techniques to be used in conjunction with a read/write policy. Rao does mention that operating parameters might include caching operations, but does not teach selective buffering as described and claimed by Applicant. Further, Rao does not teach that an application running on the processor registers itself to indicate that it complies with selective buffering techniques, or whether the process is exempt from buffering. Rao merely teaches that the operating parameters of the drive itself may be changed. These settings affect the actual drive and not an intermediate FSD which controls the buffering and read/write requests. The Examiner misunderstands the reason for registration. The Examiner submits that the reason for registration is irrelevant and must result in a structural difference. Applicant maintains that the reason is relevant and that Applicant's claimed invention is structurally different than the device taught by Rao. A user may have different read/write policies for various applications, based on expected data requests or amount of data needed. The application running on the processor registers itself with the intermediate file system driver so that the intermediate FSD can act on the application's read/write policies when the application causes a read or write request. For instance, a process could register to participate in the intermediate FSD buffering scheme for some file system requests but not others depending, for example, on state or context. In contrast, Rao teaches an interface application which just sets the operational parameters of the disk drive. This is not at all the same concept, nor does it result in application-based policies. Further, not only is the registration of an application in the intermediate FSD different than changing operating parameters of a drive, but the fact that the intermediate FSD controls the FSD based on policy and application registration is structurally different than allowing a user to select the model of a disk drive and set the operation of a file drive to access the correct type of drive.

Moreover, Claim 42, at least, is allowable as being based on an allowable base claim, as described above.

Claim rejections under 35 U.S.C. § 103

Claims 2-7 are rejected under 35 U.S.C. § 103(a) as being unpatentable over USPN 6,826,630 to Olds et al. (hereinafter, "Olds et al.") in view of USPN 6,622,252 to Klaasen et al. (hereinafter, "Klaasen et al."). This rejection is respectfully traversed and Claims 2-7 are believed allowable based on the foregoing and follow discussion.

As for Claim 2, the Examiner asserts that Olds et al. teach *if the device is determined to be inactivated, buffering the write operation to physical memory*. In fact, Olds et al. teach caching information when conditions are "favorable." This is not the same as buffering write operations to physical memory when the device is inactivated or in a limited power state.. Favorable conditions could mean anything and have any purpose. Neither a definition of "favorable" is given, nor is the purpose. A favorable condition could be that the battery power is at full level or just that the device is spun up. This would be contrary to Applicant's invention which specifically minimizes unnecessary reads/writes to the device when the device is deactivated. Further, applying the teachings of Olds et al. to Klaasen et al. will not result in Applicant's invention which optimizes power usage in a system and its devices.

Claim 2, as amended, now requires that *if the device is not operating in a limited power state then performing the write operation regardless of the active/inactive state of the device*. As described in several locations in the specification a "limited power state" is typically when the device is operating under battery power. The limited power state is irrelevant to whether the device is activated or inactivated (standby, power save mode, or spun down). Claim 2 recites that when the device is not operating under a limited power condition, that the write operation is performed regardless of the active/inactive state of the device. This is contrary to Olds et al. where the write operations are queued for sorting, etc. at all times.

Klaasen et al. teach a system for selecting modes of a drive (2-speed device) based on whether a portable device is operating under battery power. The point of their invention is to provide a lower speed using less power when under battery power. Klaasen et al. do not teach or suggest that buffering data will enhance their invention, as admitted by the Examiner. At the

cited reference, Col. 1, lines 40-57, Klaasen et al. teach disengaging the power saving mode prior to commencing read and write operations (lines 55-57). At the cited reference, Klaasen et al. teach a specific method that automatically spins the drive up to normal speed once the disk drive is initiated (lines 51-55). Klaasen et al. make no determination of the activation of the disk drive in order to select whether the write operation should be buffered. Klaasen et al. teach a method limited to the spinning up and down of the spindle to preserve battery power. However, Klaasen et al. do not teach if the device is determined to be deactivated, buffering the write operation to physical memory. A mere determination of activation and inactivation is not enough to make adding buffering obvious. Nor is it enough to make buffering data to physical memory on that condition (inactivation) obvious.

Olds et al. teach a system for reducing latency time by reordering reads/writes. Olds et al. teach a system to prioritize disk access so that caching is not necessary because read/write latency is reduced. Application of Olds et al. to Klaasen et al. would be counter-intuitive and would result in a non-optimal system for reducing power consumption as Olds et al. teaches away from buffering, by omitting a reference to the possible advantage of buffering, in order to reduce latencies. At best, combining the teaching of Klaasen et al. to Olds et al. would result in a system that buffers all commands for sorting and then performs the write commands to the disk. When no write operations are occurring, the disk is spun down to save power. If the disk is not spun up when the commands are to be executed, then the disk is spun up and the write command is committed to the disk drive. This system does not result in optimized power savings when operating under battery power (limited power state) and does not account for maintaining the write command in a buffer when the device is operating under battery power and in an inactivate state.

Claim 2 further recites that the request is received by a host processor and that the buffering is performed by an intermediate FSD executing on the processor. This is contrary to any caching performed by Olds et al. Olds et al. teach temporary caching until the disk drive spins up, or becomes "favorable." However, as shown in Fig. 2, the buffer 210 and pending command prioritization module reside on the device, i.e., like hardware caching, as illustrated by the computer 200 having an interface to the device at 202. This is contrary to Applicant's claimed invention which requires the buffering to be on memory coupled to the processor and

performed by the intermediate FSD executing on the processor. Taken either separate or apart, the cited references fail to teach or suggest all of the limitations in Claim 2; thus, Claims 2 and its progeny are believed allowable.

Claims 3-4 are believed allowable as being dependent on an allowable based claim. Whether the cited references teach systems involving disk drives, non-volatile memory components or network access drives, or a powered up/down state is irrelevant to the patenting of Claims 3-4. The cited references, either alone or in combination, do not teach or suggest all of the limitations of Claims 3-4, as dependent on Claim 2.

Claim 5 is believed allowable as being dependent on an allowable based claim. Further, as discussed above, Olds et al. teach that the caching is performed at the hardware level, on the device itself. In contrast, Claims 2 and 5 require the intermediate file system driver to be executed on the host processor and that the buffering to be in memory coupled to the host processor. In both cases, Olds et al. teach that buffering (caching), and the decision to do so, are performed by the device. Moreover, at the cited reference (Col. 2, lines 16-18), Olds et al. teach that the host computer believes the write to have actually taken place. This is contrary to Applicant`s intermediate file system driver which executes on the host processor and controls the buffering. The Examiner cites Col. 4, line 45 as an intercept operation. However, the read/write channel as taught by Olds et al. is not an intermediate file system driver, but is merely a decoder and error correction/detection circuit. Decoding and rerouting of the write request is not the same as an interception from an intermediate FSD to buffer the write operation. Thus, the cited references fail to teach the limitations of the Claim and Claim 5 is believed allowable.

As for Claim 6, the Examiner asserts that Klaasen at al. teach writing one or more buffered write operations to the device upon an occurrence of a predetermined condition and seemingly cites normal spindle velocity as the condition. (Col. 1, lines 40-57). This assertion is in error. Klaasen et al. teach that in systems of the prior art spindle speed is increased when a write request is received. Thus, when a read/write request is received by the storage device, the spindle is spun up to service the request. In contrast, Applicant's claimed invention requires writing one or more buffered write operations to the device upon occurrence of a predetermined condition and after activating the device if the device was inactivated. The Examiner obviously misunderstood the order in which activation is to take place. The activation is not the

predetermined condition. After the occurrence of the predetermined condition, the device is activated and then the buffered operation is written to the device. At the cited reference, Klaasen et al. fail to teach or suggest that one or more write operations have been buffered on the host processor memory. Klaasen et al. teach away from buffering, specifically teaching "when an access operation to the disk drive is initiated, the spindle speed is increased until the disk is rotated at the normal operating velocity. In other words, the power saving mode is disengaged prior to the commencement of read and write operations." (Col. 1, lines 51-56). It is clear that Klaasen et al. teach a system that powers down or spins down the spindles of a disk drive only until a read/write request is made, and then immediately spins up the drive. This is counter to Applicant's claimed invention which requires writing previously buffered write request. Further, the read/write requests taught by Klaasen et al. are not a *predetermined condition*, but only requests. A request is not a predetermined condition as described by Applicant. The write request is a separate operation from the buffering, which is a separate operation from the predetermined condition which is a separate operation from the activation of the device, as clearly recited in Claim 6. Even combining the teachings of the two references will not result in Applicant's claimed invention, as further discussed above. Thus, the Examiner has failed to show a *prima facie* case of obviousness and Claim 6 is believed allowable. Further, Claim 6 is allowable as being dependent on an allowable base claim.

With regards to Claim 7, as discussed above, Klaasen et al. and Olds et al. do not teach the limitations of the claim. As discussed above, none of the cited references, either alone, or in combination, teach that the request is received on a host processor and that buffering is to be performed by an intermediate FSD executing on the host processor where the predetermined condition is identified by the FSD. Thus, Claim 7 is believed allowable.

Claim 8 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Klaasen et al. and Olds et al. and further in view of USPN 6,711,686 to Barrett (hereinafter, "Barrett"). This rejection is respectfully traversed and Claim 8 is believed allowable based on the foregoing and following discussion.

As discussed above, Klaasen et al., Olds et al., do not teach all of the limitations of base Claim 7. Further, Claim 8 requires receiving user input requesting that the buffered write

operations be committed. A user input request to commit is not the same as receiving a system notification that the system is shutting down. Barrett teaches only that upon a proper "exit" from Windows-XX that the disk cache is flushed. There is no requirement in Barrett that the exit is always the result of user input. There are many case where a process, for instance a security update, might cause windows to exit. Many of these updates occur automatically, without user input. Thus, the actual teaching of Barrett is that upon a proper exit from Windows-XX, a disk cache is flushed, i.e., written. Even if the exit is caused by a user, the user has no knowledge of data being cached or previously committed. Thus, it is a misguided assumption that the exit is to be seen as equivalent to a user request to commit the buffered operations to memory. The distinction is clear. One operation is a request to exit. the claimed operation is for a user to request that the buffers be written. Further, the claimed limitation allows the user to commit buffered memory and still continue to operate, i.e., continue with the operating system running – not exited. Thus, applying Barrett to Klaasen et al. and Olds et al. will not result in Applicant's claimed invention. Thus, the Examiner has failed to show all of the limitations of Claim 8 and Claim 8 is believed allowable.

Claim 9 is rejected under 35 U.S.C. § 103(a) over Klaasen et al., Olds et al., as applied to Claim 2 and further in view of U.S. Patent Application Publication 2002/0019874 to Borr. This rejection is respectfully traversed and Claim 9 is believed allowable based on the foregoing and following discussion.

As discussed above, Klaasen et al. and Olds et al. fail to show the limitations of the base Claims. The Examiner asserts that Borr teaches *determining whether the requested write operation corresponds to an entity registered to participate in the method of controlling device write operations*. This is incorrect. Borr teaches [0107] that a file has a parameter defining its access mode, i.e., read-only, writeable, etc. The file client device, as taught by Borr, associates a lock with the file based on whether it can be written to. This is not the same as registering to participate in a method of controlling write operations, as defined by Applicants. By definition, all files in Borr *participate* in the access control method because all have an on/off flag. In contrast, Applicant requires determining whether the entity actually *participates* in the controlling process by registering with the intermediate file system driver. It will be apparent to

one of ordinary skill in the art that placing a flag in a file is not the same as proactively registering the compliance of the read/write buffering scheme with the intermediate file system driver. Applicants describe this registration scheme, at least in Para. [0017]. This registration is not a term that one of ordinary skill in the art would assume means that a flag is set in a file.

Further, Claim 9 requires that *wherein if the entity is registered with the FSD, then the buffering to be controlled based on the registration, wherein select file types are subject to selective buffering of write operations or all write operations for an application type are subject to selective buffering, and wherein write operations that are not subject to selective buffering are performed without buffering.* Borr discusses turning flags on and off for specific files - as in access control. In contrast, Applicant's claimed invention controls registration based on a file type or application type. An "entity" as recited in Claim 9 is not the same as an individual file. This term is clearly described in the specification as being associated with an application or file type. This is not the same sort of operation as described in Borr. While Applicant's invention could be used in conjunction with a file access scheme, they solve two different problems. Thus, Claim 9 is believed allowable.

Claim 10 is rejected under 35 U.S.C. § 103(a) over Klaasen et al., Olds et al., as applied to Claim 2 and further in view of USPN 5,815,648 to Giovannetti (hereinafter, "Giovannetti"). This rejection is respectfully traversed and Claim 10 is believed allowable based on the foregoing and following discussion.

As discussed above, Klaasen et al. and Olds et al. fail to teach the limitations of the claim. Therefore, combining the writeback operation of Giovannetti will not result in Applicant's claimed invention.

Claims 11-14, 16-20 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Klaasen et al. in view of Morcom. This rejection is respectfully traversed and Claims 11-14, and 16-20 are believed allowable based on the foregoing and following discussion.

As for Claims 11-14, the Examiner cannot look at each element of the recited claim in a vacuum. The power condition determines whether only the requested data is read or whether a superset of the data is read. One cannot combine the teaching of Klaasen et al. and Morcom to

provide alternate steps. Klaasen et al. specifically teach that the spindle speed is to operate at reduced rotation and/or a lower transfer rate when operating under reduced power. At no time is it suggested that that varying the amount of data to be read could be an option. Because the purpose of Klaasen et al. is to reduce the power consumption of the device by reducing the spindle rotation, applying the teaching of Morcom (to copy a maximum amount of data to cache) is counter-intuitive and would not result in Applicant's invention. Applying Morcom to Klaasen et al. would increase the power consumption by requiring excessive data to be read to fill the cache, and thus not fall within the teaching and intent of Klaasen et al. Further, Morcom teaches copying a maximum amount of data to a cache and then immediately powering down the disk regardless of the power state of the device before the read. This is counter to Applicant's claimed invention which does not immediately power down the device regardless of the power state. Thus, Morcom is misapplied to Klaasen et al. and a combination of the two references will not result in Applicant's invention.

In addition, Applicant's claim requires that the file portion is to be read into memory. The gist of the invention is to optimally access the storage device and avoid activating the device when unnecessary to minimize power consumption. Morcom teaches reading data to cache memory and not directly into physical, or system memory so that the cache is always full. This is counter to the claimed invention and also counter to the purpose of Applicant's invention. Applying Morcom would require that the cache be filled with data regardless of whether the data is logically related to the requested portion. This *relationship* is what drives whether a superset, a subset or entire file are read. Moreover, neither Klaasen et al. or Morcom teach an intermediate file system driver as recited in the amended claim, and as discussed above.

Specifically with regard to Claim 14, the Examiner attempts to introduce set theory and assert that a "subset" of the file is the same as the entire file. In light of the USPTO requirement of claim differentiation, the Examiner's interpretation cannot be the case. Claim 12 specifically requires the superset to be the entire file. Thus, as would be apparent to one of ordinary skill in the art, the subset will be understood to be a portion smaller than the entire file. Therefore this rejection is improper. Thus, Claims 11-14 are believed allowable.

As for Claim 16, Morcom teaches that the data is returned from cache memory. In contrast, Applicant's claim requires that the data be read from the device and not cache memory.

It is the requested portion that is to be returned to the requesting process (or application). the superset that was placed in the buffer remains in the buffer. Any data in the superset is maintained in the buffer, and not sent to the requesting process. Thus, applying Morcom would return data retrieved from cache memory to the process and would preclude returning data retrieved from the device. The Examiner asserts that data retrieved from cache memory is data retrieved from the device. The Examiner seems to assume that the cache memory is located on the device. However, base Claim 11 requires the buffer memory to be coupled to the processor. Thus, cache memory located on the device is not the same as the memory recited in the Claim. Thus, when the claim requires accessing the device to read the requested file portion and then returning the data to the process, there is no reading of data from the cache memory, but merely data retrieved from the device.

As for Claim 17, Morcom does not teach that the requested portion is returned to the requesting process before a remainder is read. As discussed above, Morcom does not teach a superset as defined by Applicant, and Morcom does not teach the temporal requirements of returning the requested portion first. Morcom merely teaches reading as much data as will fit in the cache. Further, the Examiner's response that it is inherent in Applicant's limitation that a superset might be read in some cases is not relevant. The limitation that is not taught or suggested by Morcom is selectively reading a superset by the intermediate FSD. Morcom reads data until the cache is full. There is no selection determination at all.

Claim 18 is allowable as being dependent on an allowable base claim, as discussed above.

As for Claim 19, Morcom does not teach or suggest *if a superset of the requested file portion is read into memory, deactivating the device*. Morcom teaches that cache is always filled to the maximum. In some cases, this might even be less than a requested portion. Further, Morcom teaches that the device is always deactivated after a read. Morcom does not teach an inherent determination that a superset of the requested portion has been read into memory or selectively choosing to read a superset by the intermediate FSD. Nor does Morcom teach that there is a possibility that the device will not be inactivated, which is inherent in Applicant's claimed invention. Moreover, Morcom teaches only that data be read into a cache and not that a requested file portion be read into memory.

Claim 20 is allowable as being dependent on an allowable base claim, as discussed above. Thus, Claims 11-14 and 16-20 are believed allowable.

Claims 15, 22-23 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Klaasen et al. and Morcom and further in view of U.S. Patent Application Publication 2002/0091902 to Hirofuji (hereinafter, "Hirofuji"). This rejection is respectfully traversed and Claims 15, 22-23 are believed allowable based on the foregoing and following discussion.

Klaasen et al. and Morcom do not teach the recited limitations, as discussed above.

As for Claim 15, Hirofuji does not teach or suggest identifying the subset of the file to be read into memory is based on one or more file access trends. Hirofuji actually discusses the arrangement of the data in the cache memory. Further, Hirofuji discusses the reading of information into cache memory in an arrangement based on likely access order, i.e., sequential or random. As described in the Specification, at least in para. [0037], Applicant describes temporal trends: i.e., which files, and which portions of files, have been accessed recently – in time. Hirofuji does not teach or suggest temporal trends, but merely characteristics of a file. Morcom teaches to read data until cache memory is full. Even if it could be surmised that Hirofuji teaches selection of data based on file access trends, it is improper to combine this with Morcom. It is improper to combine the teaching of Morcom with Hirofuji as they teach incompatible methods; thus, there is no motivation to combine, as discussed above. One cannot select a portion of a file and also fill cache until it is full. The Examiner asserts that it is possible to fill a cache with a portion of a file. However, it is not possible to do this every time, which is what would result by combining the teachings of Morcom and Hirofuji. One concept is to reduce and one concept is to increase. This result is contrary to Applicant's claimed invention. Thus, the rejection is improper and should be withdrawn.

As for Claims 22-23, Hirofuji does not teach the registering of file types with the intermediate FSD, as described by Applicant. This term is specifically discussed in the Specification in the context of Applicant's invention and other definitions cannot be assumed by one of ordinary skill in the art. Further, at the sited reference, Hirofuji only looks to see what kind of access is associated with a file (i.e., random or sequential). This is not the same as a "file type." A file type is most often associated with an extension or an application. It may be

possible for a certain file type to be accessed both randomly and sequentially, and thus the file type designation is not related to the access method. Thus, applying Hirofuji to Klaasen et al. and Morcom will not result in Applicant's claimed invention.

Also, for Claim 22, it is required that *the superset is selectively determined based in the registration*. This limitation, at least, is not taught or suggested by the cited references.

As for Claim 23, Hirofuji teaches an arrangement priority to distinguish between likely sequential or random access. This is not the same as a relative file type priority and selectively storing a superset file portion relative to the relative priority. Hirofuji teaches that data may be stored differently based on its access type. Applicant requires that a superset of the requested file potion may be selectively stored based on a relative priority. Thus, the Examiner has failed to show a *prima facie* case of obviousness and Claims 15 and 22-23 are believed allowable.

Claim 21 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Klaasen et al. and Morcom as applied to Claim 11 and further in view of USPN 6,442647 to Morton et al. (hereinafter, "Morton et al."). This rejection is respectfully traversed and Claim 21 is believed allowable based on the foregoing and following discussion.

Klaasen et al. do not disclose each and every limitation as asserted by the Examiner, as discussed above. Further, Morton et al. teach a system to take a request for data and turn it into two commands. The first request relates to the actual data request and the second relates to reading the remainder of data on the track. Morton et al. do not teach *wherein the superset of the requested file portion is logically related to the requested portion*. Morton et al. teach only that additional data on the track is read. Often this data will not be related to the requested data and especially not *logically related*. Further, the Examiner, once again, has failed to address this difference in the current Office Action. The method taught by Morton et al. is brute force to maximize a data read, but does not optimize the device access. By reading only data that is logically related to the data requested, access to the device is optimized, i.e., not over-accessed for an excessive amount of data (remainder of track). Thus, Morton et al. do not teach or suggest the recited claim limitations and Claim 21 is believed allowable.

Claims 24-26 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Olds et al. in view of USPN 5,978,921 to Ryu (hereinafter, "Ryu") in view of Klaasen et al. This rejection is respectfully traversed and Claims 24-26 are believed allowable based on the foregoing and following discussion.

As discussed above, Klaasen et al. and Olds et al. do not teach the limitations as asserted by the Examiner. Ryu teaches the automatically storing of data being used and to turn off the system in order to prevent the loss of data, i.e. hibernation. Ryu teaches detecting whether battery voltage is between a specific voltage range. Ryu teaches *selecting* a power state, i.e., power down when battery voltage is dropping. Ryu does not teach the step of *detecting* that a time-out period is about to expire and that the system is operating under a limited power condition. Ryu's device may be operating under battery power (a limited power condition as defined by Applicant), but not detect a time-out, as required in Claims 24-26. Further, Ryu teaches always storing – not buffering – data when the system battery is low. The prevention of data loss is not the same problem as optimizing power consumption and setting the device to deactivate at a specified time-out period. While Klaasen et al. teach that a spindle may be deactivated after a predetermined amount of time, there is no motivation to combine these references, as required by law. Ryu teaches a system to save data before a system is shutdown due to lack of battery power. Klaasen et al. teach a system to deactivate a spindle in a device to reduce power consumption. Neither reference suggests that its teachings could be applied to be combined with the alternate purpose. Further, neither reference teaches, *wherein the writing is in response to an intermediate file system driver detecting that a time-out has occurred.*

The Examiner asserts that Olds et al. teach detecting that a time-out period is to expire. However, the cited reference (Col. 5, line 11, col. 6, line 46 and col. 2, lines 9-11) teach or suggest no such thing. Further, the pending command prioritization module (232) of Olds et al. is not the same as an intermediate file system driver. Thus, Claim 24 and its progeny are believed allowable.

As for Claims 25-26, whether the cited references teach systems involving disk drives, non-volatile memory components or network access drives, or whether a system is operating under battery power is irrelevant to the patenting of Claims 25-26. The cited references, either alone or in combination, do not teach or suggest all of the limitations of Claims 25-26, as

dependent on Claim 24. Further, as for Claim 26, as discussed above, Ryu does not teach determining whether a limited power condition exists, but assumes battery power and just compares voltages to determine when the battery voltage is low enough to force a hibernation mode. Thus, Claims 24-26 are believed allowable.

Claims 29-30 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Klaasen et al. as applied to Claim 28 and further in view of Ryu. This rejection is respectfully traversed and Claims 29-30 and their progeny are believed allowable based on the above amendments and the foregoing and following discussion.

Claims 29-30 are, at least, allowable as being dependent on an allowable bases claim, as the cited references do not show all of the limitations in base claims 27-28. As for Claim 29, Ryu does not determine whether the device is operating under battery power. Ryu teaches only determining whether battery power for the system is equal to a specific reference voltage. Further, Ryu fails to teach writing one or more buffered write operations to the non-volatile storage device upon an occurrence of a detected predetermined condition, *wherein the predetermined condition comprises at least one condition selected from a group of conditions consisting of (i) detecting that a write buffer has become full, (ii) detecting that a certain amount of time has passed, (iii) detecting that battery power is approaching a specified threshold level, (iv) detecting that the machine is being turned off, (v) detecting that the machine is to be put in a standby state, and (vi) detecting that one of a user, a process and an operating system has explicitly requested that the write buffer contents be committed to non-volatile storage by the device*.

Ryu teaches only that data is stored in a device when battery power is low to prevent loss of data, at a specified reference voltage. Ryu does not teach detecting when a power level is *approaching* a threshold, as described in the Specification. Ryu fails to teach the other limitations of the Claim, as discussed above. Moreover, combining Ryu with the other cited references will not result in Applicant's invention. Thus, Claim 29 and its progeny are believed allowable.

As for Claim 30, the Examiner asserts the Ryu teaches *causing a machine to deactivate the non-volatile storage device after writing the one or more buffered write requests*. In fact,

Ryu teaches that the entire computer system turns itself off. This is in contrast to Applicant's invention which enables the computer system to continue to operate and only deactivates the NV storage device to save battery power. Thus, Ryu fails to teach all of the limitations of the Claim, as discussed above, either alone or in combination with the other cited references. Thus, Claim 30 and its progeny are believed allowable.

Claim 33 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Morcom as applied to Claim 31, and further in view of USPN 6,415,359 to Kimura et al. (hereinafter, "Kimura et al."). This rejection is respectfully traversed and Claim 33 is believed allowable based on the foregoing and following discussion.

Contrary to the Examiner's assertion, Morcom does not teach or suggest all of the elements of the claimed invention, as discussed in relation with Claim 31, above. Specifically, Applicant's claimed invention determines the current power state of the device and then based on that state, selectively reading a superset of the requested file portion into physical memory. Morcom reads data until the cache is full, but there is no requirement that the data is related to the requested portion, nor does Morcom teach that a lesser amount of data be read. This teaches away from selectively reading a superset. Therefore, regardless of the teaching of Kimura et al., the Examiner has failed to put forth a *prima facie* case of obviousness, as all of the limitations of the claim have not been shown by the cited references. Therefore, Claim 33 is believed allowable.

Claim 34 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Morcom in view of Morton et al. This rejection is respectfully traversed and Claim 34 is believed allowable based on the foregoing and following discussion.

Morcom does not teach the limitations of the recited Claim, as discussed above for Claim 31. Further, Morton et al. teach a system to take a request for data and turn it into two commands. The first request relates to the actual data request and the second relates to reading the remainder of data on the track. Morton et al. do not teach *wherein the superset of the requested file portion is logically related to the requested portion.* Morton et al. teach only that additional data on the track is read, as discussed above, at least for Claim 21. Therefore, the

Examiner has failed to put forth a *prima facie* case of obviousness, as all of the limitations of the claim have not been shown by the cited references. Thus, Claim 34 is believed allowable.

Claim 41 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Rao as applied to Claim 39 above, and further in view of Giovannetti. This rejection is respectfully traversed and Claim 41 is believed allowable based on the foregoing and following discussion.

Rao does not teach the limitations of the Claim, at least as discussed above in conjunction with Claim 39. At least, Rao teaches away from Applicant's claimed invention. Rao does not teach or suggest selectively buffering write requests to physical memory. Rao teaches always buffering disk writes to buffers within a SCSI controller board, not selectively buffering write request until a predetermined condition is detected. Further, neither Giovannetti nor Rao teach that write requests are intercepted and buffered by the intermediate file system driver. Therefore, a combination of the cited references will not result in Applicant's claimed invention and Claim 41 is believed allowable.

It should be noted that throughout the Office Action, the Examiner has asserted that it would be obvious to one of skill in the art to combine the references, and that to do so will result in Applicant's claimed invention. None of the asserted combinations will result in Applicant's invention or provide the advantages of such invention. Further, the Examiner continues to use the advantages of Applicant's invention to use hindsight and combine unrelated references. There has been shown no suggestion in the cited references, other prior art, or general knowledge (at the time of the invention) that there was any motivation to combine any of the references in the manner that the Examiner has done. Applicant's claimed inventions solve different problems than those in the cited references, and produce a great advantage over the prior art. It is respectfully requested that the Examiner show an actual motivation or suggestion that one of ordinary skill would be led to use one reference with another, or that all of the claims remaining in the application be permitted to issue at the earliest possible time.

CONCLUSION

In view of the foregoing, Claims 1-37, 39 and 41-42 are all in condition for allowance. If the Examiner has any questions, the Examiner is invited to contact the undersigned at (703) 633-6845. Early issuance of Notice of Allowance is respectfully requested. Please charge any shortage of fees in connection with the filing of this paper, including extension of time fees, to Deposit Account 50-0221 and please credit any excess fees to such account.

Respectfully submitted,

Dated: 5 Oct. 2007

/ Joni D. Stutman-Horn /

Joni D. Stutman-Horn, Reg. No. 42,173
Patent Attorney
Intel Corporation
(703) 633-6845

Intel Corporation
c/o Intellevate, LLC
P.O. Box 52050
Minneapolis, MN 55402